## MQTT:

MQTT (Message Queuing Telemetry Transport) is a communication protocol and data transport system. It is working as a server that transmits data between devices. It’s designed for Low-Bandwidth, High-Latency (Delay between sending and receiving). All these specifications make the MQTT ideal for IOT projects.

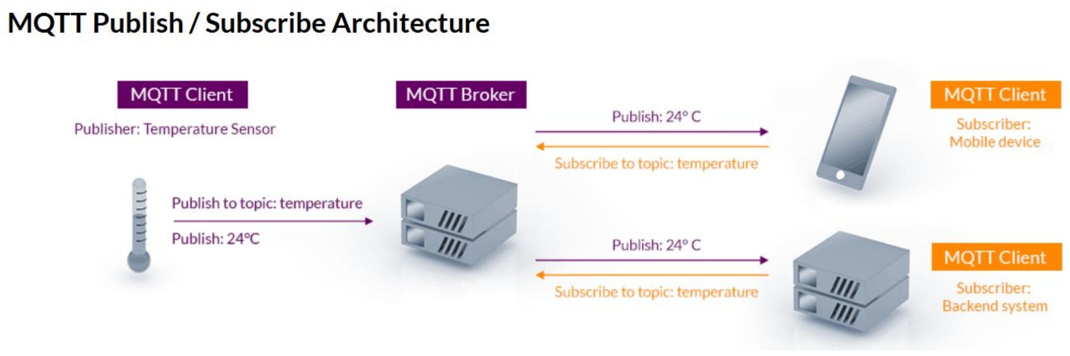
## How does it work?

It requires:

1. Server: any $\mu controller$ that sends and receive data, in our case it is ESP32.
2. Broker (server): Which is the MQTT using either HiveMQ or Mosquito.
3. Sensors: Where data is collected.
4. Chanels: it is the place where messages are exchanged.

So, at the beginning of the process for example if we are working with BMP180. The sensor will read the temperature, then the temperature will be read by the ESP32 from the sensor. Next, it will be connected to the MQTT through Wi-Fi, and finally it will send the data to a channel to display (result, graphs, etc.).

# Diagram:


MQTT Publish / Subscribe Architecture
MQTT Client
Publisher: Temperature Sensor
MQTT Broker
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Mobile device
Publish to topic: temperature
Publish: 24°C
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber: Backend system

# Comparison between MQTT using HiveMQ/Mosquito:

| Feature: | MQTT using HiveMQ | MQTT using Mosquito |
|---|---|---|
| Type: | Cloud-hosted | Local-hosted |
| Internet: | Needs Internet | Can run offline |
| Setup Complexity: | Easy | Moderate |
| UI: | Basic UI | No UI (use CLI) |
| Access: | From anywhere | Only from the Local network |
| Usage: | IOT projects, cloud projects | Local testing, full control |

So based on all this information, MQTT using HiveMQ is better for IOT and Cloud projects. So, for simplicity MQTT with HiveMQ will start first then MQTT using Mosquito.

# HiveMQ:

The MQTT data wasn't sent because HiveMQ Cloud requires a DNS-resolvable internet connection, but the iPhone hotspot didn't provide DNS or internet access to the ESP32.

Since the hotspot doesn’t act as a full router with internet routing and DNS resolution, the ESP32 couldn’t resolve the HiveMQ broker’s hostname.

The solution was to connect both the ESP32 and the PC to a local Wi-Fi network that doesn’t require a login or captive portal, allowing full device-to-device communication and DNS resolution if need.

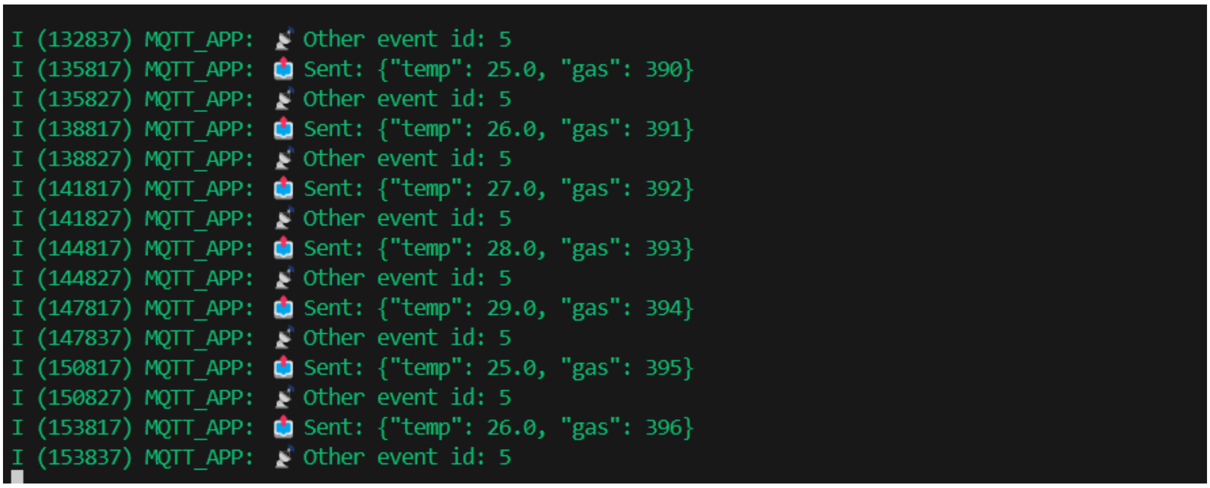
The university network will provide DNS, but the problem is that the ESP32 is not allowed to connect into these networks that ask for login.

## Mosquitto:

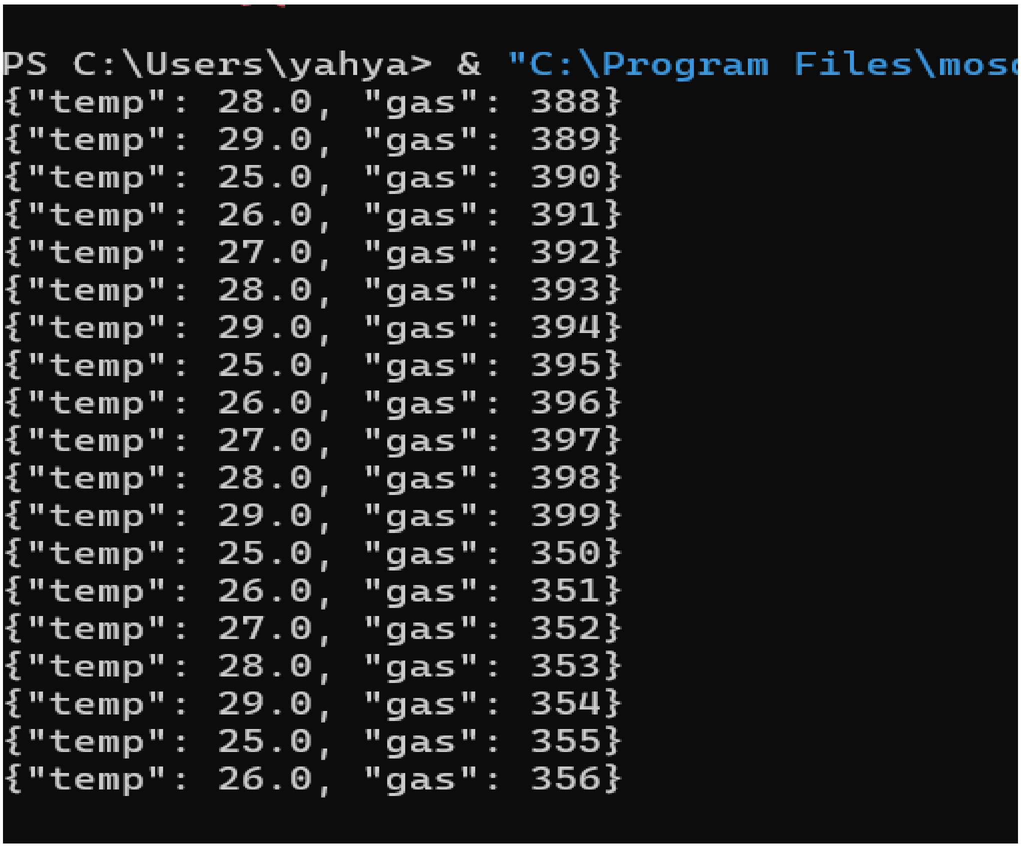
The solution was to switch to the Mosquitto MQTT broker, which works offline. It only requires that all devices are connected to the same router, so they share the same local network.

As shown in the figures, the setup worked, and I was able to successfully send data using the Mosquitto MQTT broker.

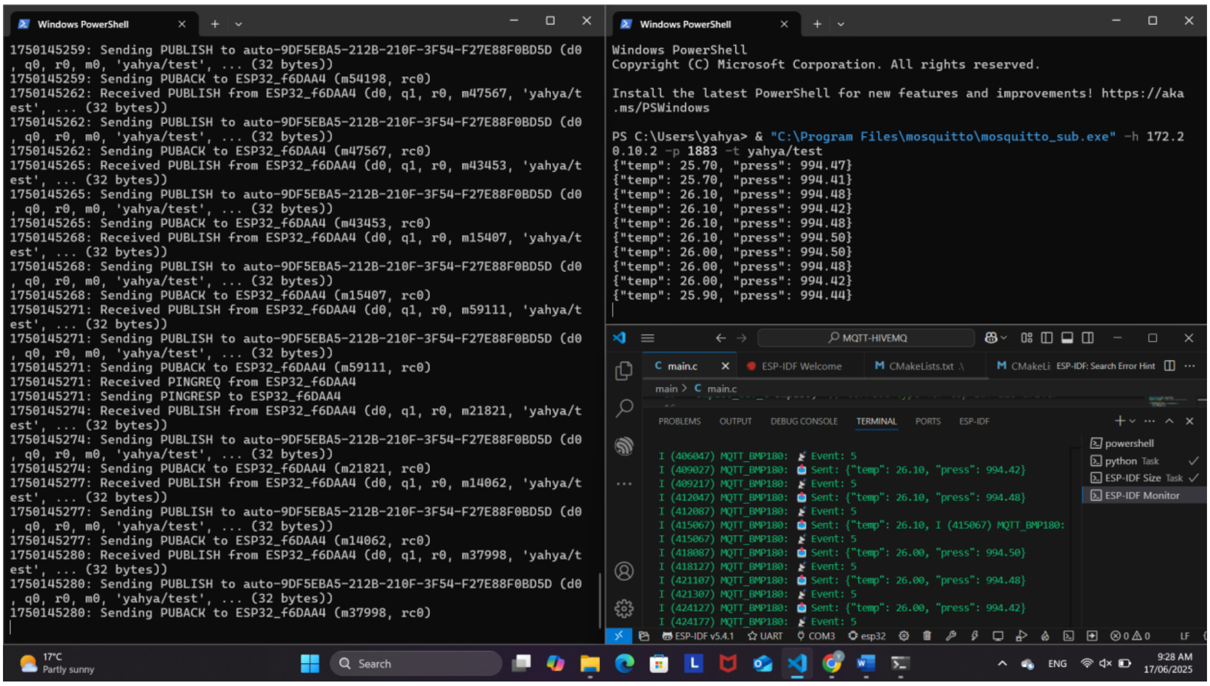
These Figures is just transmitting data between devices in the terminal.



```
I (132837) MQTT_APP: 📡 Other event id: 5
I (135817) MQTT_APP: 📤 Sent: {"temp": 25.0, "gas": 390}
I (135827) MQTT_APP: 📡 Other event id: 5
I (138817) MQTT_APP: 📤 Sent: {"temp": 26.0, "gas": 391}
I (138827) MQTT_APP: 📡 Other event id: 5
I (141817) MQTT_APP: 📤 Sent: {"temp": 27.0, "gas": 392}
I (141827) MQTT_APP: 📡 Other event id: 5
I (144817) MQTT_APP: 📤 Sent: {"temp": 28.0, "gas": 393}
I (144827) MQTT_APP: 📡 Other event id: 5
I (147817) MQTT_APP: 📤 Sent: {"temp": 29.0, "gas": 394}
I (147837) MQTT_APP: 📡 Other event id: 5
I (150817) MQTT_APP: 📤 Sent: {"temp": 25.0, "gas": 395}
I (150827) MQTT_APP: 📡 Other event id: 5
I (153817) MQTT_APP: 📤 Sent: {"temp": 26.0, "gas": 396}
I (153837) MQTT_APP: 📡 Other event id: 5
```



```
PS C:\Users\yahya> & "C:\Program Files\mosq
{"temp": 28.0, "gas": 388}
{"temp": 29.0, "gas": 389}
{"temp": 25.0, "gas": 390}
{"temp": 26.0, "gas": 391}
{"temp": 27.0, "gas": 392}
{"temp": 28.0, "gas": 393}
{"temp": 29.0, "gas": 394}
{"temp": 25.0, "gas": 395}
{"temp": 26.0, "gas": 396}
{"temp": 27.0, "gas": 397}
{"temp": 28.0, "gas": 398}
{"temp": 29.0, "gas": 399}
{"temp": 25.0, "gas": 350}
{"temp": 26.0, "gas": 351}
{"temp": 27.0, "gas": 352}
{"temp": 28.0, "gas": 353}
{"temp": 29.0, "gas": 354}
{"temp": 25.0, "gas": 355}
{"temp": 26.0, "gas": 356}
```



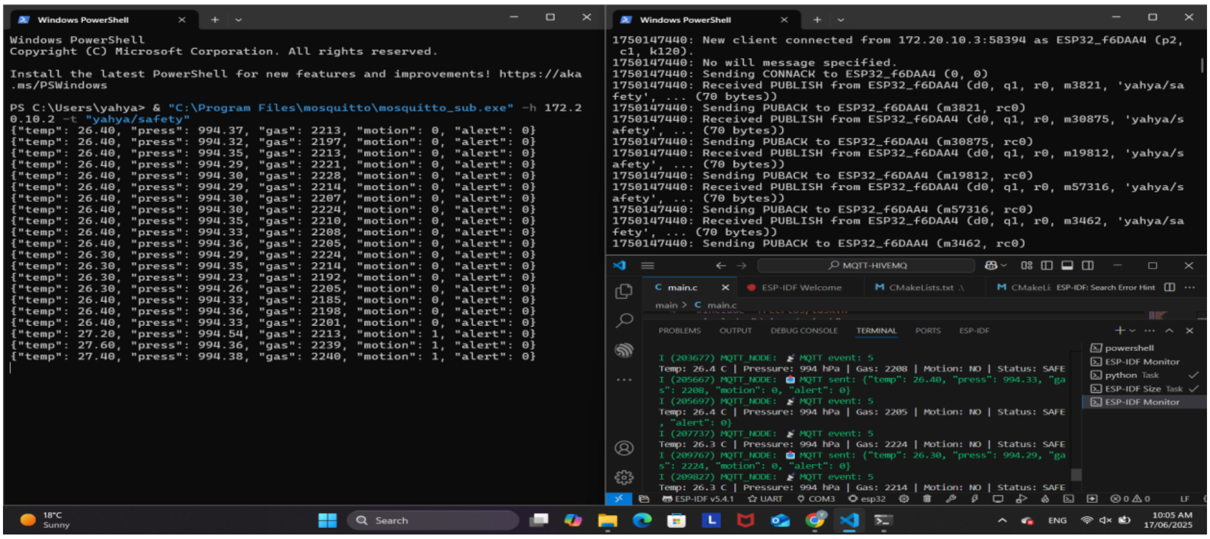
```
1750145259: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145259: Sending PUBACK to ESP32_f6DAA4 (m54198, rc0)
1750145262: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m47567, 'yahya/t
est', ... (32 bytes))
1750145262: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145262: Sending PUBACK to ESP32_f6DAA4 (m47567, rc0)
1750145265: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m43453, 'yahya/t
est', ... (32 bytes))
1750145265: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145265: Sending PUBACK to ESP32_f6DAA4 (m43453, rc0)
1750145268: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m15407, 'yahya/t
est', ... (32 bytes))
1750145268: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145268: Sending PUBACK to ESP32_f6DAA4 (m15407, rc0)
1750145271: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m59111, 'yahya/t
est', ... (32 bytes))
1750145271: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145271: Sending PUBACK to ESP32_f6DAA4 (m59111, rc0)
1750145271: Received PINGREQ from ESP32_f6DAA4
1750145271: Sending PINGRESP to ESP32_f6DAA4
1750145274: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m21821, 'yahya/t
est', ... (32 bytes))
1750145274: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145274: Sending PUBACK to ESP32_f6DAA4 (m21821, rc0)
1750145277: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m14062, 'yahya/t
est', ... (32 bytes))
1750145277: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145277: Sending PUBACK to ESP32_f6DAA4 (m14062, rc0)
1750145280: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m37998, 'yahya/t
est', ... (32 bytes))
1750145280: Sending PUBLISH to auto-9DF5EBA5-212B-210F-3F54-F27E88F0BD5D (d0
, q0, r0, m0, 'yahya/test', ... (32 bytes))
1750145280: Sending PUBACK to ESP32_f6DAA4 (m37998, rc0)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka
.ms/PSWindows

PS C:\Users\yahya> & "C:\Program Files\mosquitto\mosquitto_sub.exe" -h 172.2
0.10.2 -p 1883 -t yahya/test
{"temp": 25.70, "press": 994.47}
{"temp": 25.70, "press": 994.41}
{"temp": 26.10, "press": 994.48}
{"temp": 26.10, "press": 994.42}
{"temp": 26.10, "press": 994.48}
{"temp": 26.10, "press": 994.50}
{"temp": 26.00, "press": 994.50}
{"temp": 26.00, "press": 994.48}
{"temp": 26.00, "press": 994.42}
{"temp": 25.90, "press": 994.44}
```

```
I (406047) MQTT_BMP180: Event: 5
I (409027) MQTT_BMP180: Sent: {"temp": 26.10, "press": 994.42}
I (409217) MQTT_BMP180: Event: 5
I (412047) MQTT_BMP180: Sent: {"temp": 26.10, "press": 994.48}
I (412087) MQTT_BMP180: Event: 5
I (415067) MQTT_BMP180: Sent: {"temp": 26.10, I (415067) MQTT_BMP180:
I (415067) MQTT_BMP180: Event: 5
I (418087) MQTT_BMP180: Sent: {"temp": 26.00, "press": 994.50}
I (418127) MQTT_BMP180: Event: 5
I (421107) MQTT_BMP180: Sent: {"temp": 26.00, "press": 994.48}
I (421307) MQTT_BMP180: Event: 5
I (424127) MQTT_BMP180: Sent: {"temp": 26.00, "press": 994.42}
I (424177) MQTT_BMP180: Event: 5
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka
.ms/PSWindows

PS C:\Users\yahya> & "C:\Program Files\mosquitto\mosquitto_sub.exe" -h 172.2
0.10.2 -t "yahya/safety"
{"temp": 26.40, "press": 994.37, "gas": 2213, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.32, "gas": 2197, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.35, "gas": 2213, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.29, "gas": 2221, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.30, "gas": 2228, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.29, "gas": 2214, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.30, "gas": 2207, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.30, "gas": 2224, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.35, "gas": 2210, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.33, "gas": 2208, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.36, "gas": 2205, "motion": 0, "alert": 0}
{"temp": 26.30, "press": 994.29, "gas": 2224, "motion": 0, "alert": 0}
{"temp": 26.30, "press": 994.35, "gas": 2214, "motion": 0, "alert": 0}
{"temp": 26.30, "press": 994.23, "gas": 2192, "motion": 0, "alert": 0}
{"temp": 26.30, "press": 994.26, "gas": 2205, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.33, "gas": 2185, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.36, "gas": 2198, "motion": 0, "alert": 0}
{"temp": 26.40, "press": 994.33, "gas": 2201, "motion": 0, "alert": 0}
{"temp": 27.20, "press": 994.54, "gas": 2213, "motion": 1, "alert": 0}
{"temp": 27.60, "press": 994.36, "gas": 2239, "motion": 1, "alert": 0}
{"temp": 27.40, "press": 994.38, "gas": 2240, "motion": 1, "alert": 0}
```

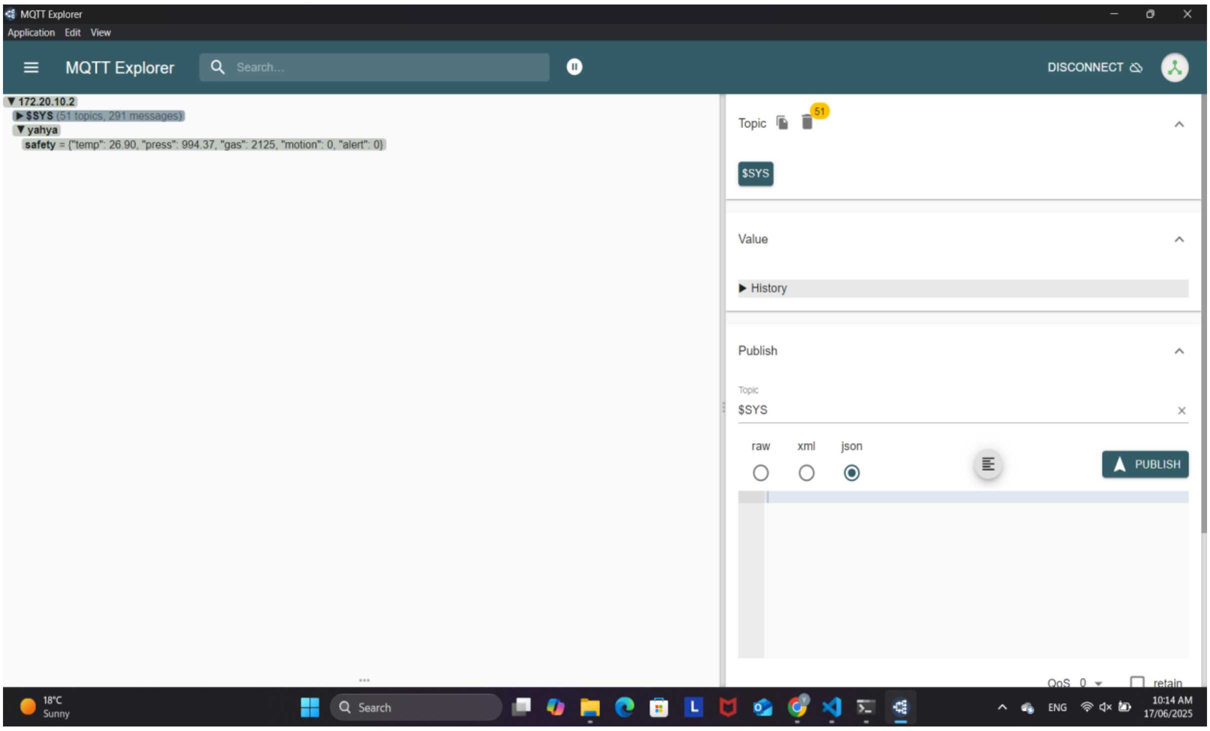
```
1750147440: New client connected from 172.20.10.3:58394 as ESP32_f6DAA4 (p2,
 c1, k120).
1750147440: No will message specified.
1750147440: Sending CONNACK to ESP32_f6DAA4 (0, 0)
1750147440: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m3821, 'yahya/sa
fety', ... (70 bytes))
1750147440: Sending PUBACK to ESP32_f6DAA4 (m3821, rc0)
1750147440: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m30875, 'yahya/s
afety', ... (70 bytes))
1750147440: Sending PUBACK to ESP32_f6DAA4 (m30875, rc0)
1750147440: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m19812, 'yahya/s
afety', ... (70 bytes))
1750147440: Sending PUBACK to ESP32_f6DAA4 (m19812, rc0)
1750147440: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m57316, 'yahya/s
afety', ... (70 bytes))
1750147440: Sending PUBACK to ESP32_f6DAA4 (m57316, rc0)
1750147440: Received PUBLISH from ESP32_f6DAA4 (d0, q1, r0, m3462, 'yahya/sa
fety', ... (70 bytes))
1750147440: Sending PUBACK to ESP32_f6DAA4 (m3462, rc0)
```

```
I (203677) MQTT_NODE: MQTT event: 5
Temp: 26.4 C | Pressure: 994 hPa | Gas: 2208 | Motion: NO | Status: SAFE
I (205667) MQTT_NODE: MQTT sent: {"temp": 26.40, "press": 994.33, "ga
s": 2208, "motion": 0, "alert": 0}
I (205697) MQTT_NODE: MQTT event: 5
Temp: 26.4 C | Pressure: 994 hPa | Gas: 2205 | Motion: NO | Status: SAFE
, "alert": 0}
I (207737) MQTT_NODE: MQTT event: 5
Temp: 26.3 C | Pressure: 994 hPa | Gas: 2224 | Motion: NO | Status: SAFE
I (209767) MQTT_NODE: MQTT sent: {"temp": 26.30, "press": 994.29, "ga
s": 2224, "motion": 0, "alert": 0}
I (209827) MQTT_NODE: MQTT event: 5
Temp: 26.3 C | Pressure: 994 hPa | Gas: 2214 | Motion: NO | Status: SAFE
```

As explained above, MQTT Mosquitto is not a cloud service; it is a local broker used to transmit data between devices. To send this data to the cloud, tools like MQTT Explorer and Node-RED can be used as interfaces or bridges to cloud platforms.

# MQQT Explorer:

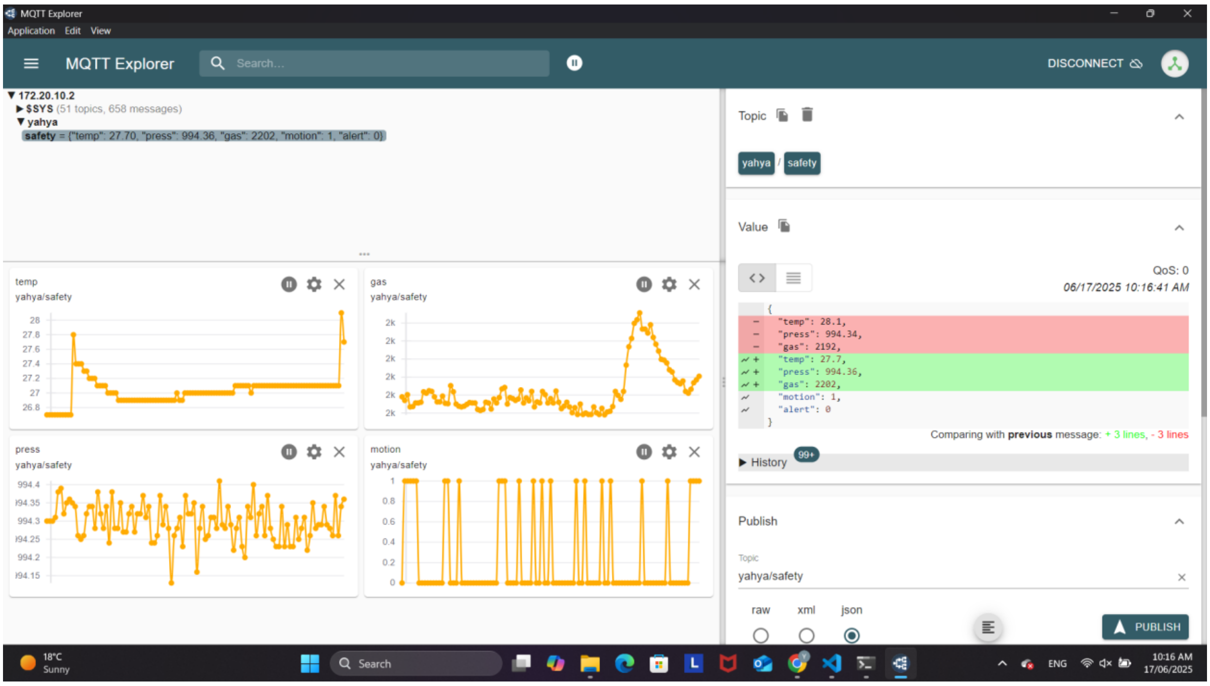
MQTT Explorer is a powerful desktop application used for monitoring and debugging MQTT communication. It provides a user-friendly graphical interface that allows users to connect to an MQTT broker, subscribe to topics, publish messages, and view real-time data flow. MQTT Explorer is especially useful for testing and validating MQTT-based systems, as it displays topic hierarchies, payloads, QoS levels, and timestamps in an organized and interactive format. It supports both local and cloud brokers, making it a versatile tool for developers working with IoT devices and messaging systems.



```
▼ 172.20.10.2
  ▶ $SYS (51 topics, 291 messages)
  ▼ yahya
    safety = {"temp": 26.90, "press": 994.37, "gas": 2125, "motion": 0, "alert": 0}
```

Topic: $SYS

Publish Topic: $SYS



```
▼ 172.20.10.2
  ▶ $SYS (51 topics, 658 messages)
  ▼ yahya
    safety = {"temp": 27.70, "press": 994.36, "gas": 2202, "motion": 1, "alert": 0}
```

Topic: yahya / safety

QoS: 0
06/17/2025 10:16:41 AM

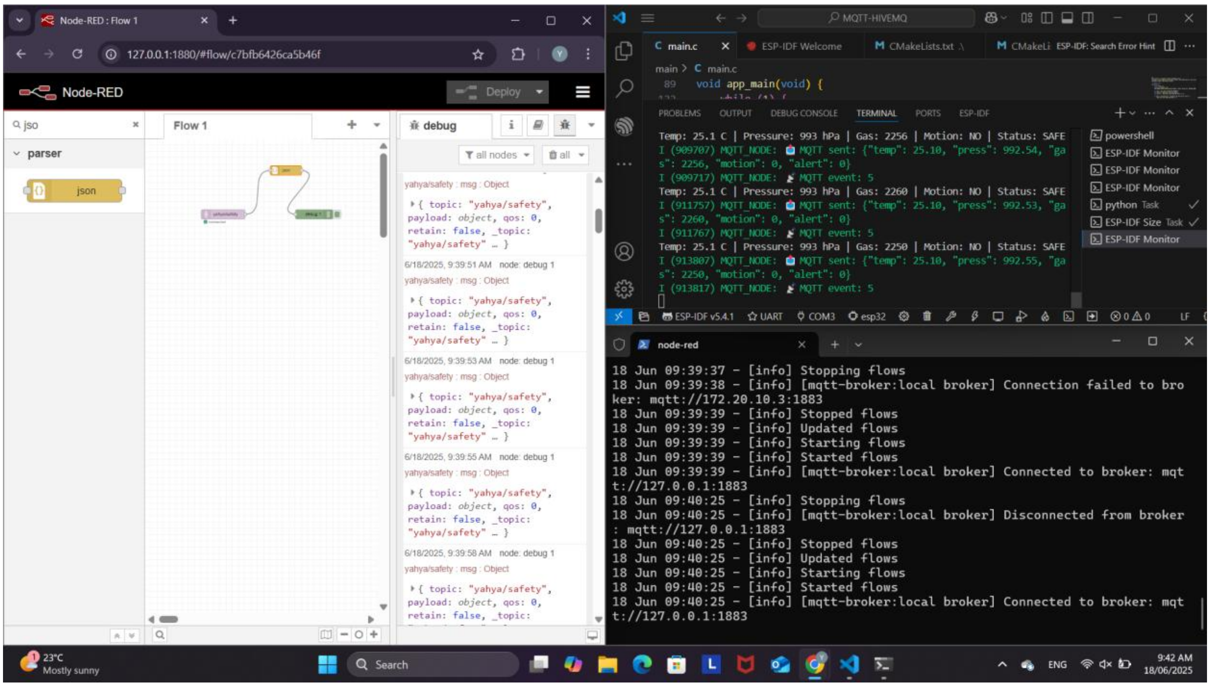
```
    {
 -    "temp": 28.1,
 -    "press": 994.34,
 -    "gas": 2192,
 +    "temp": 27.7,
 +    "press": 994.36,
 +    "gas": 2202,
      "motion": 1,
      "alert": 0
    }
```

Comparing with **previous** message: + 3 lines, - 3 lines

Publish Topic: yahya/safety

# Node-Red:

Node-RED is a visual programming tool used for connecting hardware devices, APIs, and online services in a simple, flow-based interface. It allows users to create automation workflows by wiring together nodes that represent different functions, such as MQTT input/output, data processing, and cloud integration. Node-RED is especially popular in IoT projects because it makes it easy to collect, process, and visualize data from sensors and devices without writing complex code. It runs on platforms like PCs, Raspberry Pi, and cloud servers, offering flexibility and real-time interaction.

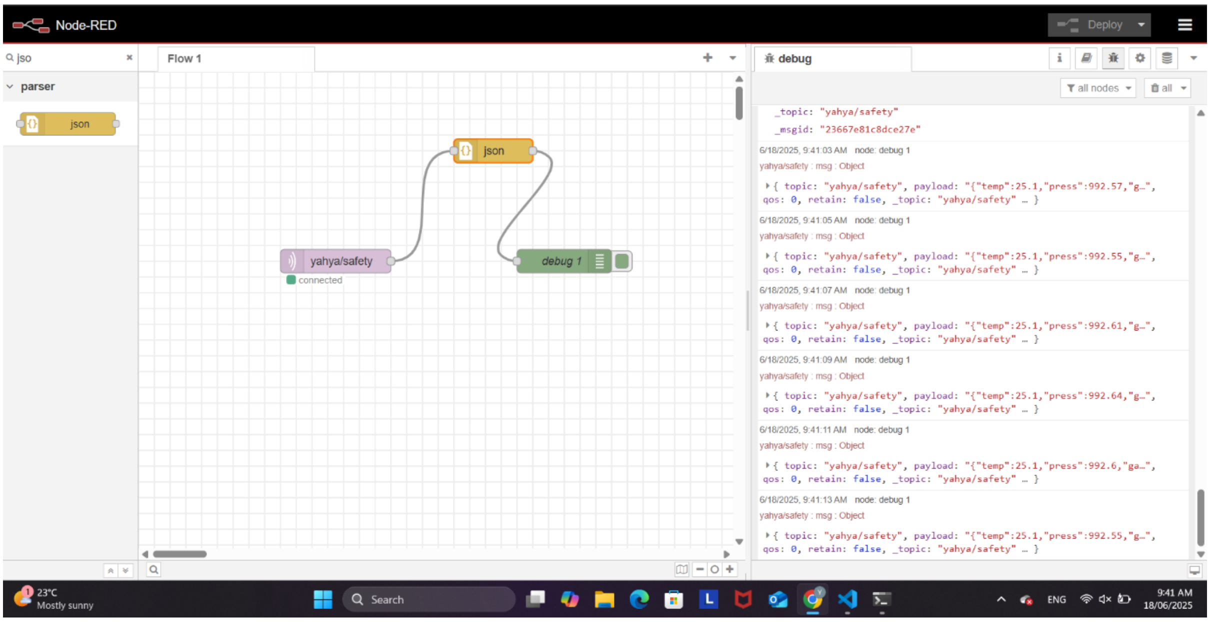


```
yahya/safety : msg : Object
▸{ topic: "yahya/safety",
payload: object, qos: 0,
retain: false, _topic:
"yahya/safety" … }
6/18/2025, 9:39:51 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety",
payload: object, qos: 0,
retain: false, _topic:
"yahya/safety" … }
6/18/2025, 9:39:53 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety",
payload: object, qos: 0,
retain: false, _topic:
"yahya/safety" … }
6/18/2025, 9:39:55 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety",
payload: object, qos: 0,
retain: false, _topic:
"yahya/safety" … }
6/18/2025, 9:39:58 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety",
payload: object, qos: 0,
retain: false, _topic:
```

```
void app_main(void) {
```

```
Temp: 25.1 C | Pressure: 993 hPa | Gas: 2256 | Motion: NO | Status: SAFE
I (909707) MQTT_NODE: 📤 MQTT sent: {"temp": 25.10, "press": 992.54, "ga
s": 2256, "motion": 0, "alert": 0}
I (909717) MQTT_NODE: 📡 MQTT event: 5
Temp: 25.1 C | Pressure: 993 hPa | Gas: 2260 | Motion: NO | Status: SAFE
I (911757) MQTT_NODE: 📤 MQTT sent: {"temp": 25.10, "press": 992.53, "ga
s": 2260, "motion": 0, "alert": 0}
I (911767) MQTT_NODE: 📡 MQTT event: 5
Temp: 25.1 C | Pressure: 993 hPa | Gas: 2250 | Motion: NO | Status: SAFE
I (913807) MQTT_NODE: 📤 MQTT sent: {"temp": 25.10, "press": 992.55, "ga
s": 2250, "motion": 0, "alert": 0}
I (913817) MQTT_NODE: 📡 MQTT event: 5
```

```
18 Jun 09:39:37 - [info] Stopping flows
18 Jun 09:39:38 - [info] [mqtt-broker:local broker] Connection failed to bro
ker: mqtt://172.20.10.3:1883
18 Jun 09:39:39 - [info] Stopped flows
18 Jun 09:39:39 - [info] Updated flows
18 Jun 09:39:39 - [info] Starting flows
18 Jun 09:39:39 - [info] Started flows
18 Jun 09:39:39 - [info] [mqtt-broker:local broker] Connected to broker: mqt
t://127.0.0.1:1883
18 Jun 09:40:25 - [info] Stopping flows
18 Jun 09:40:25 - [info] [mqtt-broker:local broker] Disconnected from broker
: mqtt://127.0.0.1:1883
18 Jun 09:40:25 - [info] Stopped flows
18 Jun 09:40:25 - [info] Updated flows
18 Jun 09:40:25 - [info] Starting flows
18 Jun 09:40:25 - [info] Started flows
18 Jun 09:40:25 - [info] [mqtt-broker:local broker] Connected to broker: mqt
t://127.0.0.1:1883
```



```
  _topic: "yahya/safety"
  _msgid: "23667e81c8dce27e"
6/18/2025, 9:41:03 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.57,"g…",
qos: 0, retain: false, _topic: "yahya/safety" … }
6/18/2025, 9:41:05 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.55,"g…",
qos: 0, retain: false, _topic: "yahya/safety" … }
6/18/2025, 9:41:07 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.61,"g…",
qos: 0, retain: false, _topic: "yahya/safety" … }
6/18/2025, 9:41:09 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.64,"g…",
qos: 0, retain: false, _topic: "yahya/safety" … }
6/18/2025, 9:41:11 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.6,"ga…",
qos: 0, retain: false, _topic: "yahya/safety" … }
6/18/2025, 9:41:13 AM  node: debug 1
yahya/safety : msg : Object
▸{ topic: "yahya/safety", payload: "{"temp":25.1,"press":992.55,"g…",
qos: 0, retain: false, _topic: "yahya/safety" … }
```

## MQTT Explorer & Node-Red:

Both are powerful tools. MQTT Explorer is mainly used for visualizing MQTT messages in a structured format, making it ideal for monitoring and debugging. However, for creating graphs and interactive dashboards, Node-RED is a better choice due to its built-in visualization and flow automation features.

## Resources:

https://mqtt.org/

https://www.cloudamqp.com/docs/mqtt.html?utm_source=google&utm_medium=cpc&utm_campaign=19661297433&utm_term=mqtt%20protocol&gad_source=1&gad_campaignid=19661297433&gbraid=0AAAAApKbGlUdauSbkgPshE6Wjno0QFmy&gclid=CjwKCAjwgb_CBhBMEiwA0p3oOBtaMoSDL25PcFP1VEF9Zc5oolGfmUc_phGHJ9hwtgjAT5nz7UiUwRoCddkQAvD_BwE

https://mosquitto.org/

https://www.hivemq.com/